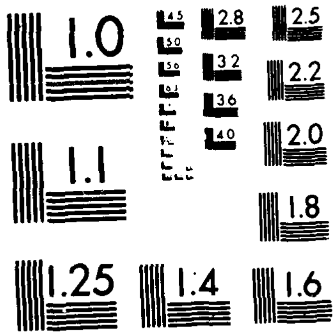
AD-A143 433 EXPERIMENTAL EVALUATION OF FIVE TECHNIQUES FOR TEACHING FOR THE ZOG FRAME..(U) CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE C K ROBERTSON ET AL. 1/1

UNCLASSIFIED 14 JUN 84 CMU-CS-84-140 F33615-78-C-1551 F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

CMU-CS-84-140

AD-A143 433

Experimental Evaluation of Five Techniques for Teaching for the ZOG Frame Editor

14 June 1984

C. Kamila Robertson
Allen Newell

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

CMU-CS-84-140

Experimental Evaluation of Five Techniques for Teaching for the ZOG Frame Editor

14 June 1984

C. Kamila Robertson
Allen Newell

Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA, USA

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory under Contract F33615-78-C-1551. It was also partially supported by the Office of Naval Research under Contract N00014-76-0874, and by a grant from the Palo Alto Research Center of Xerox Corporation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Office of Naval Research, Xerox Corporation, or the U. S. Government.

Table of Contents

A-1

List of Figures

EXPERIMENTAL EVALUATION OF FIVE TECHNIQUES FOR TEACHING FOR THE ZOG FRAME EDITOR

Abstract

With the growing number of computer text editing systems, a question arises about the best way to teach beginners to use a given text editor. We are particularly concerned with choosing among several teaching techniques that are available for users of Carnegie-Mellon University's ZOG system and its editor, ZED. (ZOG is a rapid-response, menu-selection, software system intended as a general-purpose interface to a computer.) This paper compares five techniques for teaching naive users to edit using ZED: three forms of a manual (*on-line* in ZOG's net-structured format, *off-line*, and *on-line* displayed in parallel with the *on-line* editing work); a highly structured *tutorial*; and a *human teacher*. The techniques are compared with each other and with eight editors evaluated by Roberts and Moran (1982). The results indicate (1) all the techniques take essentially the same time to produce adequate learning; (2) the style in which the teaching techniques are used varies according to accessibility and structure; and (3) ZED learning falls in the middle of the range of Roberts and Moran's editors in terms of minutes required on average to learn to do a new editing task.

1. INTRODUCTION

In the past few years there has been a growing interest in evaluating human-computer interfaces, including interfaces to computer text editors. Several studies (Card, Moran & Newell, 1983, Robertson, C. Kamila, McCracken & Newell, 1981) model users' interaction with an editor in terms of keystrokes and time required to acquire the next unit of text modification. Roberts (1979) and Roberts and Moran (1982, 1983) applied this model to compare time to learn a basic core of editor commands for eight editors — TECO, WYLBUR, NLS, WANG, BRAVOX, BRAVO, GYPSY, and EMACS.

ZOG is an interactive system developed at Carnegie-Mellon University (Robertson, G., McCracken & Newell, 1981). The ZOG project has a goal of responding rapidly to users' difficulties and continuously improving the system. Therefore it is especially important to evaluate ZOG in the context of real users doing real tasks in a real computer usage environment. A particular concern is to evaluate various teaching and help mechanisms which are already in use. They must be robust, i.e., allow the user to avoid or recover from errors. They must provide effective and accessible information about the system, even though the system is undergoing frequent design changes. Of particular importance is the question of the style of teaching that will be most effective. We are especially concerned with beginners, but the resources available to beginners are available also to experts for reference. For instance, is a user consultant or human teacher needed or can the user (even the beginner) really teach himself? We would also like to evaluate users' performance in ZOG's editor ZED (as taught in these various ways) with respect to the behavior of beginners using other systems. ZED combines facilities like those of other editors with facilities specialized to the hierarchical character of ZOG's databases. In a previous paper (Robertson, C. Kamila, McCracken & Newell, 1981), we studied time for experts to complete a standard set of editing tasks using ZED. For those studies, Roberts' editor evaluation scheme (Roberts, 1979) offered the possibility of relatively straightforward comparison of ZED with other editors.

In this paper we look at the behavior of beginners learning ZED,[1] measured by time to learn how to perform a basic set of editing tasks. This measure will be used to evaluate five teaching techniques that are now in use by beginners (and accessible to experts) learning ZED. We continue to use Roberts' experimental

[1] Preliminary results were reported in (Robertson, C. Kamila & Akscyn, 1982).

scheme as a framework for comparison. Although we use an experimentally structured task and environment to assess how well users learn ZED, what is to be learned is not an abstracted or abbreviated part of learning to edit. ZOG (and ZED) are in daily use both in research and applied environments. The major example of the latter is an installation aboard the nuclear aircraft carrier USS CARL VINSON, in which a distributed system of some 25 ZOG workstations is being used for management operations (Newell et al, 1982). The variations to be evaluated here each constitute the candidate learning experience for the user learning ZED. Afterwards the user is essentially on his own to acquire the additional commands provided by the editor.

Below, we first present a brief description of ZOG. Following this, we describe our experiment with beginners. Then we discuss the differences among the teaching techniques with respect to Roberts' measures. We also analyze the style in which the teaching techniques are actually used. Finally, we discuss the results of comparing overall ZED learning with that of eight other editors studied by Roberts and Moran.

2. THE SYSTEM CONTEXT: ZOG

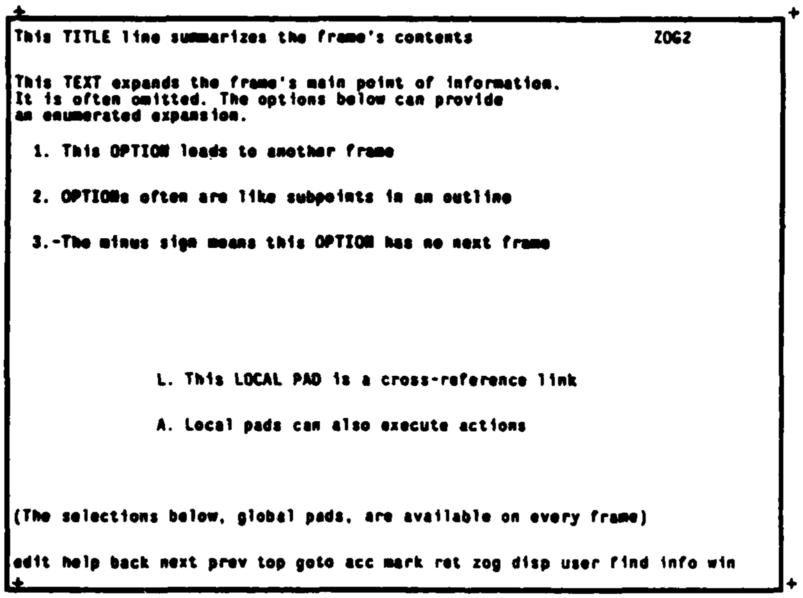


Figure 2-1: A self-describing ZOG frame.

ZOG is a general purpose, rapid response, menu-selection interface to a computer system. ZOG's databases are strongly hierarchical, multiply linked nets of displays called *frames*, each the size of a conventional standard (24 X 80) terminal display screen[2]. Each frame (see Figure 2-1) consists of a set of *items*: a title; a

---

[2] One condition uses a recently developed *two-window* ZOG, which runs on a PERQ personal computer with a larger display screen. Two ZOG frames appear at once, one above the other, e.g., for parallel searching, or for reference in one area of the net while editing in another.

few lines of text: a set of numbered (or lettered) menu items called *options* and *local pads*; and a line of ZOG commands called *global pads* at the bottom of the screen. Global pads include *back* (back up one frame) and *edit* (edit the current frame). An option, local pad, or global pad is selected by a single character (usually the initial character of its display), which then displays another frame and/or executes a program. ZOG nets range from tens to tens of thousands of frames.

ZED is a display editor with commands for editing the textual content of the frame, rearranging the positions of items on the frame, and editing the non-displayed information such as next-frame links. Most ZED commands are single characters. After the user has selected the global pad *edit*, all keyboard input is interpreted as ZED commands rather than ZOG selections. Within ZED there are several modes: *command* mode, in which characters are interpreted as commands and command arguments; *insert* mode, in which characters are inserted into the text at the current cursor location; *position-item* mode, in which the cursor is used to change the position of an item in the display; and *help* mode, in which the user traverses a net of help frames. The *exit* command returns the user to ZOG selection mode.

3. EXPERIMENTAL DESIGN

The question posed in this experiment is: for ZOG-naive users, how long will it take to learn the basic core of ZED commands with different techniques for delivering information about ZED? Specifically, we consider: (1) *human teacher*, (2) on-line *tutorial*, (3) *on-line* manual, (4) *two-window* on-line manual, and (5) *off-line* manual. All of the schemes are available in our environment and are under consideration as possible standard teaching techniques. In particular, they have been under consideration as techniques for teaching ZOG/ZED to the mixed user population of the USS CARL VINSON.

The teaching technique is the independent variable. Each teaching technique provides the same content. The techniques differ chiefly in the way the user accesses them (by searching on-line or by page turning, for example), and in who controls the access (the user or a teacher). *Tutorial* and *teacher* conditions mandate the size and distribution of instruction, whereas the forms of the manual do not. Also, the *two-window* and *on-line* manuals (being nets) are hierarchically organized, so that the user has an overview of the topics at some level, unless he is actually reading the instruction on a particular command or concept. *On-line* and *two-window* manuals always place the user at the top of the net when first accessed. The *off-line* manual contains a table of contents, but the user is not forced to read it.

Learning time is the dependent variable. The total learning effort is composed of a set of tasks, and the learning time is indexed by the average time to learn these tasks.

To compare learning scores for ZED with other editors, calculated by Roberts' (1979) method, calibration is provided by running one additional condition with a human teacher teaching the EMACS editor, replicating one of Roberts and Moran's conditions. We can then compare *teacher*-EMACS with *teacher*-ZED, and *teacher*-ZED with the other ZED teaching techniques.

The complete design is shown in Figure 3-1. Part I is a preparatory step to learn the basic ZOG system, which does did not involve editing with ZED. It consists of instruction in basic ZOG searching, followed by a game for practice. The time for Part I was limited to about an hour, based on our experience of previous beginners learning ZOG searching. Acquiring basic skill in searching a ZOG net is a prerequisite to using ZED. (It also provides a useful measure of how much time users actually needed for the basic ZOG instruction.)

Part II is the main learning experiment and corresponds to Roberts' method (described below). During the learning experiment (using Roberts' stimuli), users are taught 23 basic tasks. Quizzes (the mandatory part of the assigned editing) contain 49 instances of those basic tasks.

Part III — henceforth called the *retention test* — is a test administered without any of the sources of instruction, about a week after Parts I and II, to check for long-term learning. Retention-test tasks were created to have the same structure as Roberts' exercises and quizzes. The user does 22 tasks (i.e., instances of basic tasks). He then must correct mistakes and omissions found by the experimenter. Time to total completion is then recorded.

Part I	Tutorial style introduction to ZOG net searching					
Part II	On-line Manual	Tutorial	Two-Win. Manual	EMACS Teacher	Off-line Manual	ZED Teacher
cycle1	net search Quiz1	tutorial section Quiz1	net search Quiz1	oral lesson Quiz1	document search Quiz1	oral lesson Quiz1
cycle2	lesson Quiz2	lesson Quiz2	lesson Quiz2	lesson Quiz2	lesson Quiz2	lesson Quiz2
...						
cycle5	lesson Quiz5	lesson Quiz5	lesson Quiz5	lesson Quiz5	lesson Quiz5	lesson Quiz5
Part III	Editing test: memo, autobiography, science fiction selections done to completion (all correct)					

Figure 3-1: Design of the ZED learning experiment.

4. METHODOLOGY

Roberts and Moran were interested in variations over editors, using a fixed teaching technique. In contrast, we are interested in variations in learning a single editor, due to teaching technique. However, Roberts' method has proved highly applicable to our goals. She developed a set of experiments including a test of time to learn a set of commonly used *core* commands, a score card for functionality, a test of expert performance time, and a score card for error and disaster potential. For this experiment, we used her learning paradigm, which follows a set syllabus. The syllabus introduces a set of basic editing tasks with a sequence of exercises, each followed by a quiz. Exercises are optional; the user is to do as much and as many as he feels he needs to learn to use the editor. The mandatory quizzes provide an opportunity to assess learning. A short summary of commands is available throughout, in hardcopy.

Roberts developed a set of *core tasks*, which are those that all editors can perform and are also the most common editing tasks in normal applications. A *task* consists of finding the next editing change in the (hardcopy) manuscript, locating the change in the version on the system, modifying the text, and (optionally) verifying the change. A task comprises an operation (inserting, deleting, replacing, moving, splitting, or merging) and an operand (character, word, line, sentence, or paragraph). The teaching sequence is composed of a set of five alternating exercises and quizzes. Each of these is composed of a set of editing tasks (covering 23 out of a total of 39 defined tasks). The tasks are indicated by corrections, marked in red on the hardcopy, which the user makes to the online copy.

During quizzes the user is to ask questions and use the summary only if absolutely necessary. Quizzes are scored cumulatively. The user receives one point for each task that was done correctly on a quiz (by whatever method), and done correctly on subsequent quizzes if there was opportunity. This constitutes learning the task. The principal data collected, besides quiz scores, are: (1) total task time, and (2) overall time per task learned (not all users will do the same number of tasks because of possible omissions). Roberts provides a fixed set of quizzes and exercises to teach and test these tasks. The user is assigned a learning score, in minutes per task learned. The average of all users' scores is the score for the editor. In this experiment, the average score for all users in a given condition is the score for that teaching technique.

In agreement with the philosophy of Roberts and Moran, we have used a minimum number of subjects per cell (four). This is based on balancing the effort of data collection with the need to achieve statistical significance in the data comparisons. Four is the smallest number which allows for both. This implies that only relatively strong differences will emerge clearly, a feature that is appropriate in applied contexts.

5. PROCEDURE

5.1. Users

Users were four beginners per condition. A beginner is defined as a college student or equivalent who has had at least one session on a terminal, but no more than one computing course or the equivalent. In this experiment, we found that most of the students who applied to be our users had some (less than one year) experience with EMACS, a display oriented editor in extensive use at Carnegie-Mellon. EMACS has a set of commands that is very different from ZED. Thus our users had had some editing experience, but with a set of commands that would not transfer directly to ZED use.[3]

5.2. The Task

Roberts' documents were mapped onto ZOG frames, with approximately 10 to 12 lines of text per frame. Frames in the exercises and quizzes were linked linearly (that is, with a minimum of hierarchical structure) to minimize ZOG searching. The core of editing tasks in ZED was defined so that editing was done within a fixed net structure. Tasks included moving text between frames using the move/copy facility, but not changing the basic net structure. Most ZED editing in fact occurs within frames, and the editors with which ZED was being compared contain nothing comparable to net building. This task is realistic for ZOG use and

[3] It is almost impossible at CMU to find a student with no computer experience, since the students teach each other, and since they are encouraged to use the machine at least to write their papers; however, we did find the required number of EMACS novices for the EMACS condition.

is similar to the ongoing training situation of people learning to use ZOG/ZED at present.

Roberts' syllabus had to be adapted to work with all of our teaching techniques. For the *off-line* condition, the ZED manual chapter on editing, plus the entire table of contents, was available. The user could look up something specific or just read the manual.[4] The *off-line* manual was presented in a three-ring notebook. The *on-line* and *two-window* manuals were contained in ZOG nets. *On-line* was accessed by a local pad ("M. Manual") from every introduction, exercise, and quiz frame. The *two-window* manual was accessed via the h ("help") ZED command. It appeared in whichever window was not being used for editing. The *on-line* and *two-window* manuals consisted of the same text as the *off-line* manual, one concept or command usually corresponding to one frame. The user searched the manual net and then used a global pad to return directly to the frame from which he started, outside the manual. The *human teacher* and *tutorial* instruction sequences followed Roberts' syllabus as closely as possible. However, in all conditions search by content was learned early, although in the syllabus this comes at the end. ZED editing depends heavily on the user's ability to search by content.

A copy of the document net was created for each user to modify. One user at a time sat at a PERQ (personal computer) display simulating a Concept terminal, with a 9600 baud hardwired line to a DEC Vax 11/780 computer. ZOG was already invoked, and the appropriate teaching technique was ready. Each user was given a single teaching technique.

The rule for questions and use of the summary was as in Roberts and Moran's method: any time during exercises, but only if absolutely necessary during quizzes. In addition, during quizzes, the user was to limit his use of the teaching technique (e.g., the manual) to occasions when he was unable to continue otherwise. ZED help frames could be used at any time. (In the *two-window* condition, help evoked the manual itself; there was no other on-line ZED help.)

5.3. Data Collection and Treatment

Each user was videotaped. A copy of the screen display the user was reading was superimposed on the television picture, along with a millisecond timestamp. Videotape data were accurate to one thirtieth of a second (the frequency of the video frames). During the session, ZOG unobtrusively recorded the user's path through the net and the selections and editing commands at each frame, each timestamped, on a log file. The data were pooled to provide information on the users' editing style and learning progress.

Part II time was partitioned with respect to the various possible activities in which the user could be engaged, with the purpose of characterizing learning style. Users could spend their time as follows: (1) ZED *learning*, composed of reading the manual or listening to the teacher, using ZED help, using the summary sheet and asking questions; (2) editing *performance*, composed of studying the stimulus (task) sheets and actual editing; and (3) *other*, composed of reading or listening to instructions about the experiment, taking breaks, etc. *Total time* is composed of *learning* and *editing*. *Total time* was divided by the cumulative quiz score to obtain the learning score.

For the retention test, the initial time and total time to completion (after corrections) were observed. Significant non-editing delays were removed from these figures.

The teaching techniques were characterized and compared for the amount and distribution of use of the teaching technique and of the other teaching aids (questions, help, and summary sheet). Quiz scores were obtained by comparing hardcopy of the edited frames with the quiz documents.

Part I time does not enter into ZED learning, but we are interested in the length of time it takes to learn

[4] The complete *off-line* manual is *The ZOG User's Guide* (Yoder & Akscyn, 1982).

basic ZOG searching as well. Hence we collected the time to do the initial ZOG search instruction, and the total Part I time which also included searching practice in the form of a game. *Two-window* Part I included additional instruction on searching with the *two-window* screen.

5.4. Expectations for User Behavior

The five teaching techniques differ in the way information is available about what is to be learned. We would expect user behavior to be influenced by a variety of factors, such as the following.

- *Sequence* — Whether the sequence in which the material is to be learned is *variable* (controlled by the user), *fixed* (controlled by ZOG), or *semifixed* (controlled by the teacher responding to the user's needs).
- *Control* — Whether the *user*, the *teacher*, or the *ZOG* system (ols access to the information.
- *Structure* — Whether the information is structured *linearly* (ook, on *demand* in answer to questions, in a *fixed* sequence of frames, or *hierarchically* in a . always starting at the top of the net).
- *Access* — Whether the desired information is only *indirectly* available after traversing additional material or whether it is *directly* available.
- *Context* — Whether, when information is obtained, the current editing context is *lost* or *retained*.
- *Movement* — Whether access to information requires *no* physical movement or a major movement of *turning* the body away from the editing posture.

The following table shows the values taken by the different teaching techniques on each factor.

Condition	On-line	Off-line	Teacher	Tutorial	TwoWindow
Sequence	Variable	Variable	Semifixed	Fixed	Variable
Control	User	User	Teacher	ZOG	User
Structure	Hierarchy	Linear	Demand	Fixed	Hierarchy
Access	Indirect	Direct	Direct	Direct	Indirect
Context	Lost	Retained	Retained	Retained	Retained
Movement	No	Turn	Turn	No	No

As the table shows, each technique has a unique set of values. There are many possible trade-offs and we cannot tell in advance how these factors will balance out. To add to the complexity, computer manuals are often seen as difficult to understand. They might be expected to be the least effective from the point of view of content. We might also predict that the *human teacher* will be fastest. A question from the user may elicit information that the user would not have known to request. This increases the effect of direct access to information. The *tutorial* will probably be the slowest, since there is a fixed learning sequence and exercises are required at every step.

Why work with stimuli which differ along so many dimensions? The experiment is not designed around carefully constructed psychological stimuli, but rather around real teaching techniques in use in the ZOG computing environment. Several major pedagogical alternatives are represented here, and we wish to know the effectiveness of those alternatives. To clean up the teaching techniques to make them easier to study would remove their direct relevance — a not uncommon dilemma in applied research.

6. RESULTS

In the Part II (learning) segment of the experiment, the EMACS condition indicates whether our results are comparable with Roberts and Moran's. A major difference in learning score between our EMACS users' average learning score and that of their EMACS users would tell us to be cautious. Our EMACS users averaged 6.5 minutes per task; theirs averaged 6.6. (Both sets of EMACS users are represented in Figure 6-2 and Figure 6-6, which will be discussed below.) These averages are close enough to indicate that our experiment is generally comparable with theirs and that we can place our ZED users in Roberts' and Moran's continuum of editors.

The basic results of the experiment are shown in Figure 6-1. The figures are mean times with the coefficient of variation (CV, the standard deviation divided by the mean) in parentheses after the mean. Percentages are labeled.

Condition	On-line	Off-line	Teacher	Tutorial	TwoWindow	Emacs
Basic ZOG Searching	25.58 (.57)	21.66 (.08)	19.89 (.16)	29.36 (.28)	47.98 (.28)	32.35 (.50)
Part I Total Time	53.68 (.20)	52.01 (.12)	57.85 (.11)	53.73 (.24)	61.90 (.11)	59.08 (.23)
Learning Scores	7.44 (.17)	6.52 (.26)	6.05 (.27)	7.10 (.19)	6.80 (.28)	6.48 (.41)
Part II Totals	162.57 (.18)	149.35 (.24)	136.55 (.27)	159.21 (.20)	147.28 (.18)	145.06 (.41)
Learning time	29.48 (.34)	38.16 (.37)	40.00 (.43)	32.45 (.09)	24.96 (.61)	47.50 (.24)
Performance time	124.97 (.18)	109.50 (.35)	94.75 (.28)	122.92 (.27)	105.85 (.14)	96.46 (.51)
Learning + Performance (LP)	154.45 (.17)	147.66 (.28)	134.75 (.20)	155.38 (.76)	130.81 (.19)	143.96 (.40)
Learning % of LP	19%	26%	30%	21%	19%	33%
Performance % of LP	81%	74%	70%	79%	81%	67%
Quiz 5, Time per task	2.50 (.91)	1.30 (.31)	1.60 (.37)	1.60 (.35)	1.50 (.42)	1.50 (.59)
Retention test, Time per task	1.90 (.23)	1.30 (.06)	1.30 (.25)	1.50 (.45)	1.30 (.10)	1.96 (.59)
Initial time	35.53 (.21)	26.14 (.05)	28.57 (.27)	32.15 (.42)	27.29 (.14)	38.61 (.58)
Correction time	7.57 (1.6)	2.57 (1.0)	.22 (1.2)	.83 (1.4)	.45 (1.2)	.77 (1.5)
Total time	43.10 (.33)	28.71 (.07)	28.79 (.26)	32.98 (.45)	27.75 (.12)	38.97 (.59)

Figure 6-1: Results of the learning experiment.

Our learning-score results are compared with Roberts and Moran's in Figure 6-2. The left column contains scores for the ZED teaching techniques; the right column contains learning scores for Roberts and Moran's editors. T-tests show no significant differences among the scores for ZED. The mean of the five ZED learning scores is 6.8 minutes per task. Means are given plus or minus the coefficient of variation (CV).

ZED Learning Scores		Roberts and Moran's Learning Scores	
		Teco	19.5 $\pm$ .29
		Wylbur	8.2 $\pm$ .24
		NLS	7.7 $\pm$ .26
On-Line	7.4 $\pm$ .17		
		Bravo	7.3 $\pm$ .14
Tutorial	7.1 $\pm$ .19		
TwoWindow	6.8 $\pm$ .28		
		Emacs	6.6 $\pm$ .22
Emacs	6.5 $\pm$ .41		
Off-Line	6.5 $\pm$ .26		
		Wang	6.2 $\pm$ .45
Teacher (ZED)	6.1 $\pm$ .27		
		Bravox	5.4 $\pm$ .08
		Gypsy	4.3 $\pm$ .26

Figure 6-2: Learning scores (min $\pm$ CV).

Learning curves for our users and for Roberts and Moran's users are represented in Figure 6-3, in the same (somewhat conventional) format as Roberts' (1979) Figure 4.1. Our Figure 6-3 also contains plots of Roberts' data for her worst editor, TECO. The TECO curve is for the better time to learn, which Roberts obtained with her second teacher. For comparison, WANG was the best editor of Roberts' 1979 experiment; the curve for WANG falls almost on top of our *on-* and *off-line* curves. However, its total time is just over 120 minutes, compared with times ranging from roughly 120 to 160 minutes for our five conditions. (See Figure 6-2 for a minutes per task comparison.) The sloping segments of each curve represent time spent in instruction and exercises. The horizontal segments represent quiz time. This format represents the user's knowledge as increasing during non-quiz time and remaining constant during quizzes, but realistically, some learning does occur during quizzes.

Total Part II (ZED teaching) times correspond with the endpoints of the curves and are given in Figure 6-1. T-tests show no significant differences among teaching techniques for ZED. The closeness of the curves confirms this.

Quiz 5 is the point at which the users had gone through the entire instruction sequence, before the one-

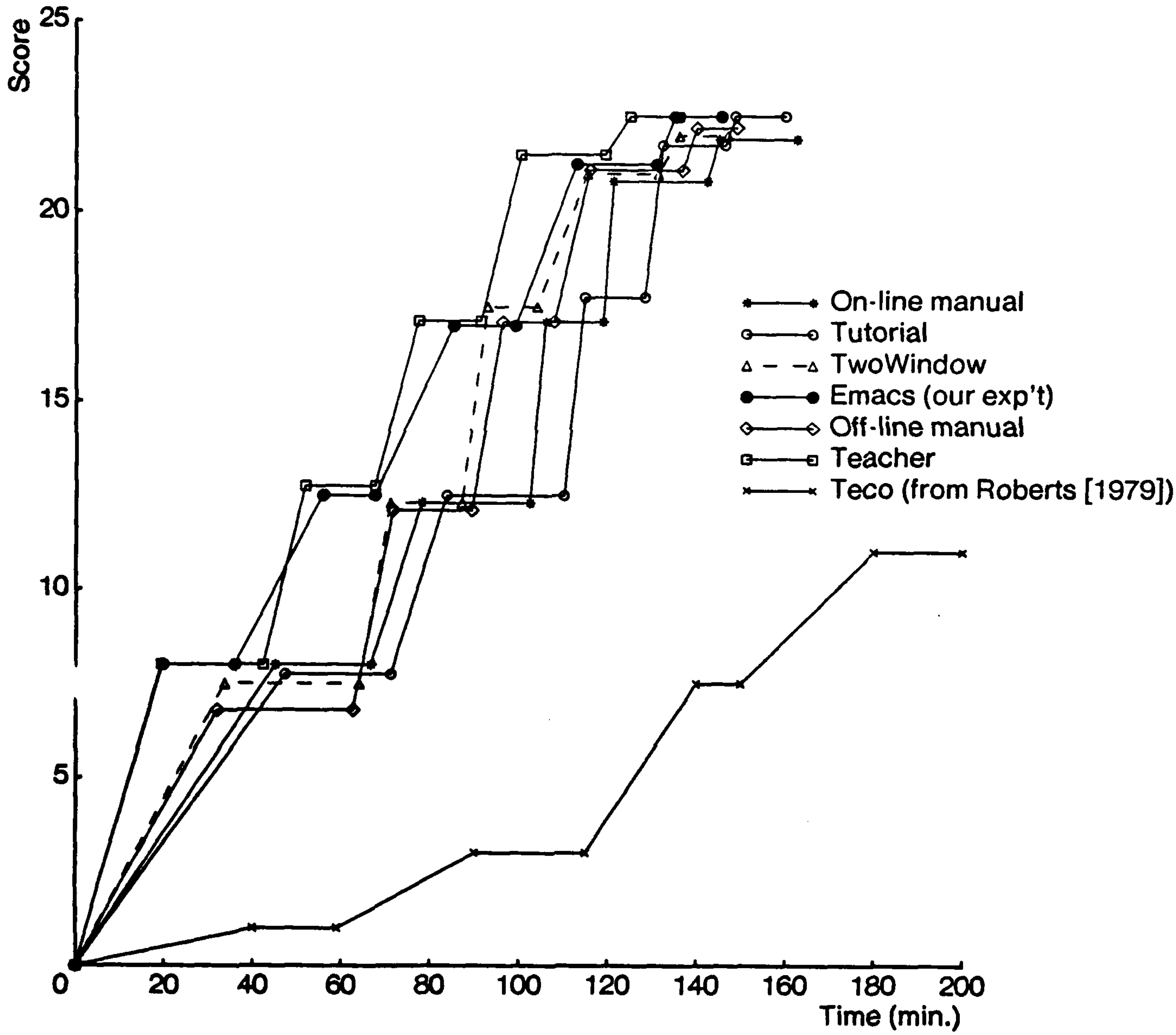


Figure 6-3: Average learning curves.

week wait for the retention test. T-tests comparing Quiz 5 time per task for the various conditions show no significant differences.

In the retention test, total time to completion is composed of: (1) initial time (until the user first said he was finished); and (2) correction time if any (time to correct mistakes and omissions discovered by the experimenter). The results are shown graphically in Figure 6-4. T-tests indicate that the *on-line* initial time was significantly greater than *off-line* (α = .05). This is the only significant difference in this category. It should be noted that in the *on-line* condition, one user's retention test initial time was much higher than the others' — 1.4 standard deviations above the mean and 1.3 standard deviations above the next highest initial time.

Using Figure 6-4, we can compare the retention test with Quiz 5. That is, we are testing whether the week-long interval between Part II and the retention test made a difference in retention of ZED editing skills. The Quiz 5 times shown are minutes per task for seven tasks; the retention test times are minutes per task for 22 tasks. Users took slightly more time per task for the seven tasks immediately after instruction, than for the 22 tasks a week later. T-tests comparing Quiz 5 with the retention test for each condition indicate that there are no significant differences. Thus the users performed as well a week later. Given that the expectation is for a decline in performance after a week, this establishes that the learning was effective.

6.1. Learning and Performance

In the learning and performance partitions of users' time, by far the largest percentage of all time spent in Part II was spent in the editing aspect of performance (60 to 70%). The next largest partition was the manual reading (teacher) aspect of learning (10 to 23%). Figure 6-5 shows the mean number of minutes spent in performance and in learning, with instruction and editing times in the bottom segment of each bar. (In our discussion of learning and performance, "instruction time" means time spent receiving ZED instruction, whether from the teacher, a manual, or the tutorial.)

T-tests indicate no significant differences among the learning times or in performance times. However, there were two significant differences in the instruction portion of learning time: *on-line* was less than *off-line* and *tutorial* at $\alpha = .05$. Use of other aids (ZED help, questions, and the summary sheet) was a very small percentage of the learning time in all conditions.

6.2. ZED Compared with Other Editors

Figure 6-6 plots learning scores and mean scores for all of our conditions, along with those of Roberts and Moran [after Figure 4 of (Roberts & Moran, 1982)]. Our two EMACS users have been represented in the same column with Roberts and Moran's EMACS users, to show the degree to which our two EMACS conditions had similar results. ZED users appear in the middle of the range of Roberts and Moran's editors. Roberts (1979) gives data for individual users for four of the editors, so we can compare them with our users statistically. [Only graphical data was available in (Roberts & Moran, 1982) for the other four.] All ZED groups had significantly better learning scores than Roberts' faster set of TECO users, who learned from Roberts' second teacher ($\alpha = .005$ for all). T-tests comparing the ZED conditions with Roberts' other editors (WYLBUR, NLS, and WANG) do not show significant results. Roberts' tests indicate that all her TECO users had significantly higher (worse) learning scores than users of her three other editors, and there were no significant differences among the three editors. Our results place all ZED users in the faster of the two overall groups in Roberts' 1979 study.

All of the users are of course novices. On average, they took between 1.23 and 1.40 minutes per task (retention test initial and total times respectively). For comparison, expert ZED users take about .50 minutes per task (Robertson, C. Kamila, McCracken & Newell, 1981). Roberts and Moran's expert users took about .82 minutes per task for TECO (the slowest), .62 for EMACS, and .32 for GYPSY (the fastest).

6.3. Basic ZOG Learning

To study the learning of the basic ZOG system, which is essentially learning to search ZOG nets, we calculated Part I time, and also time spent in the initial searching segment of Part I. (The latter segment of Part I was searching practice.) The time in the initial segment ranged from 19 to 32 minutes, except *two-window* (48 minutes). T-tests show that *two-window* time for the basic ZOG instruction is significantly greater than for all other conditions, at $\alpha = .01$ (except *two-window* vs. *on-line*, $\alpha = .025$). We attribute this difference to the additional instruction on *two-window* searching, which occurred only in that condition. There were no other significant differences for initial ZOG instruction time. Average initial time for all users

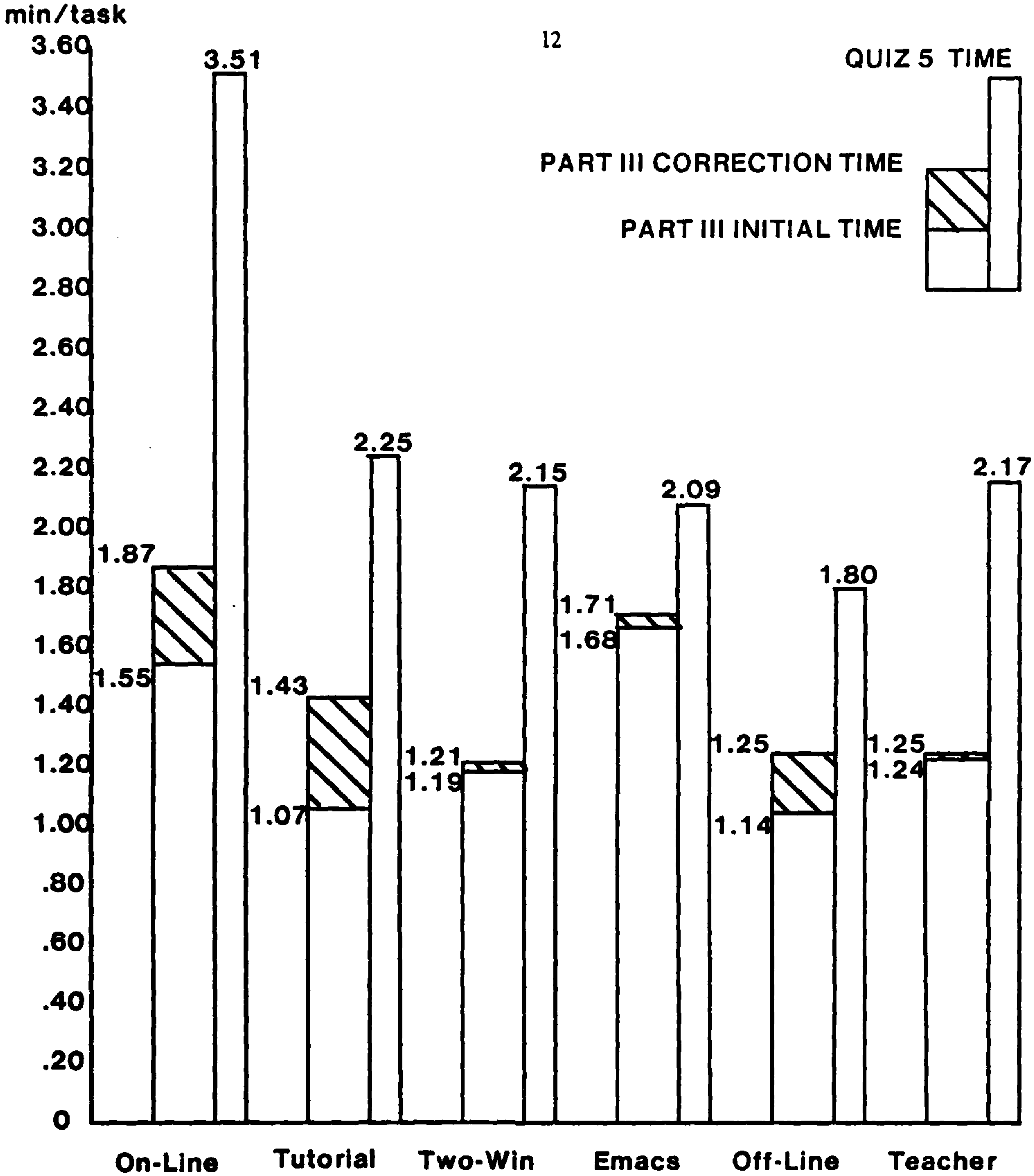


Figure 6-4: Quiz 5 compared with the retention test, time per task.

other than *two-window* was 24 minutes (CV = .31).

Total Part I time ranged from 49 to 61 minutes. (Users could leave Part I when they were finished, so some took less than the hour that was scheduled; nevertheless, differences among the conditions were not statistically significant.)

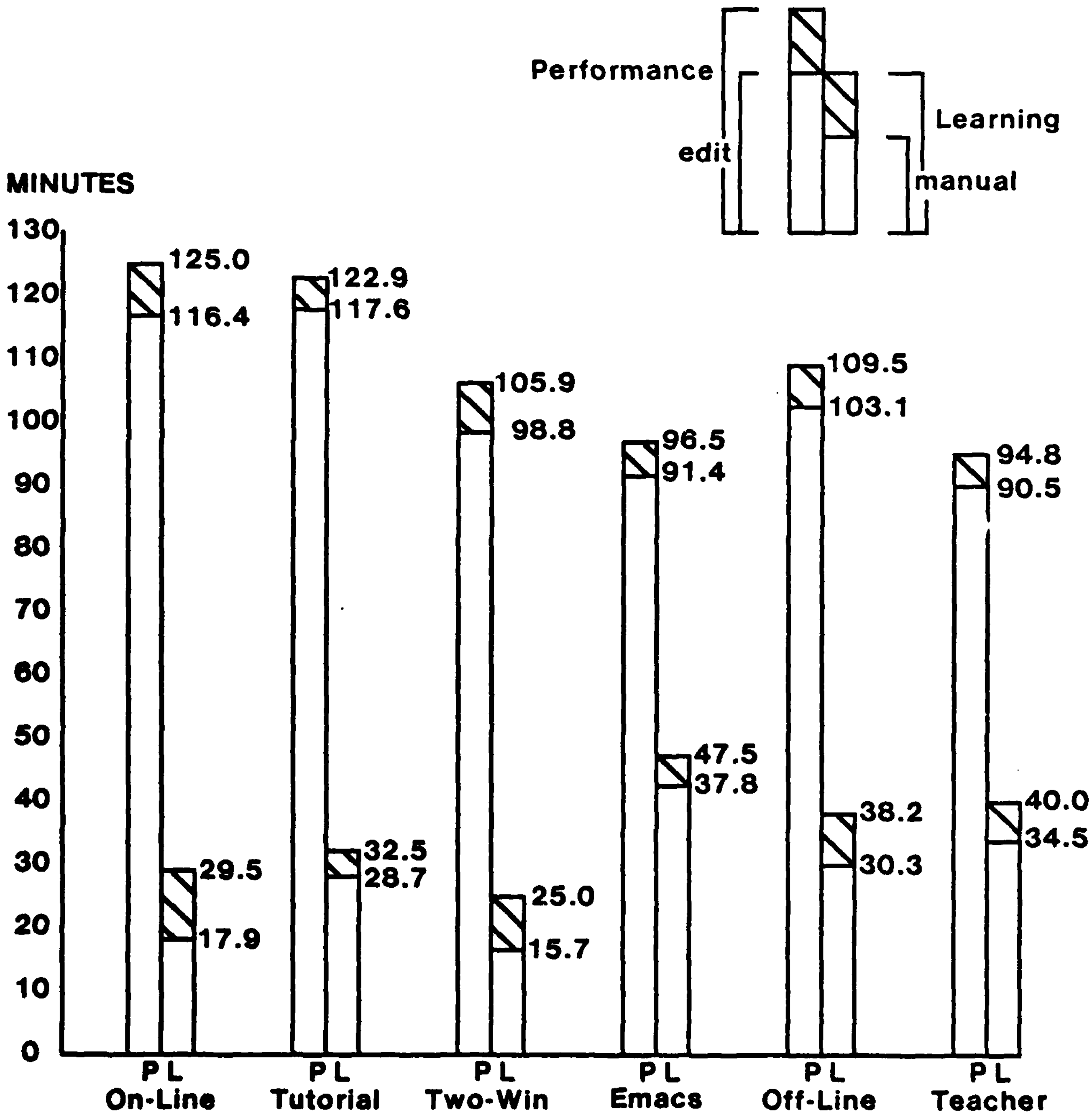


Figure 6-5: Learning and performance times.

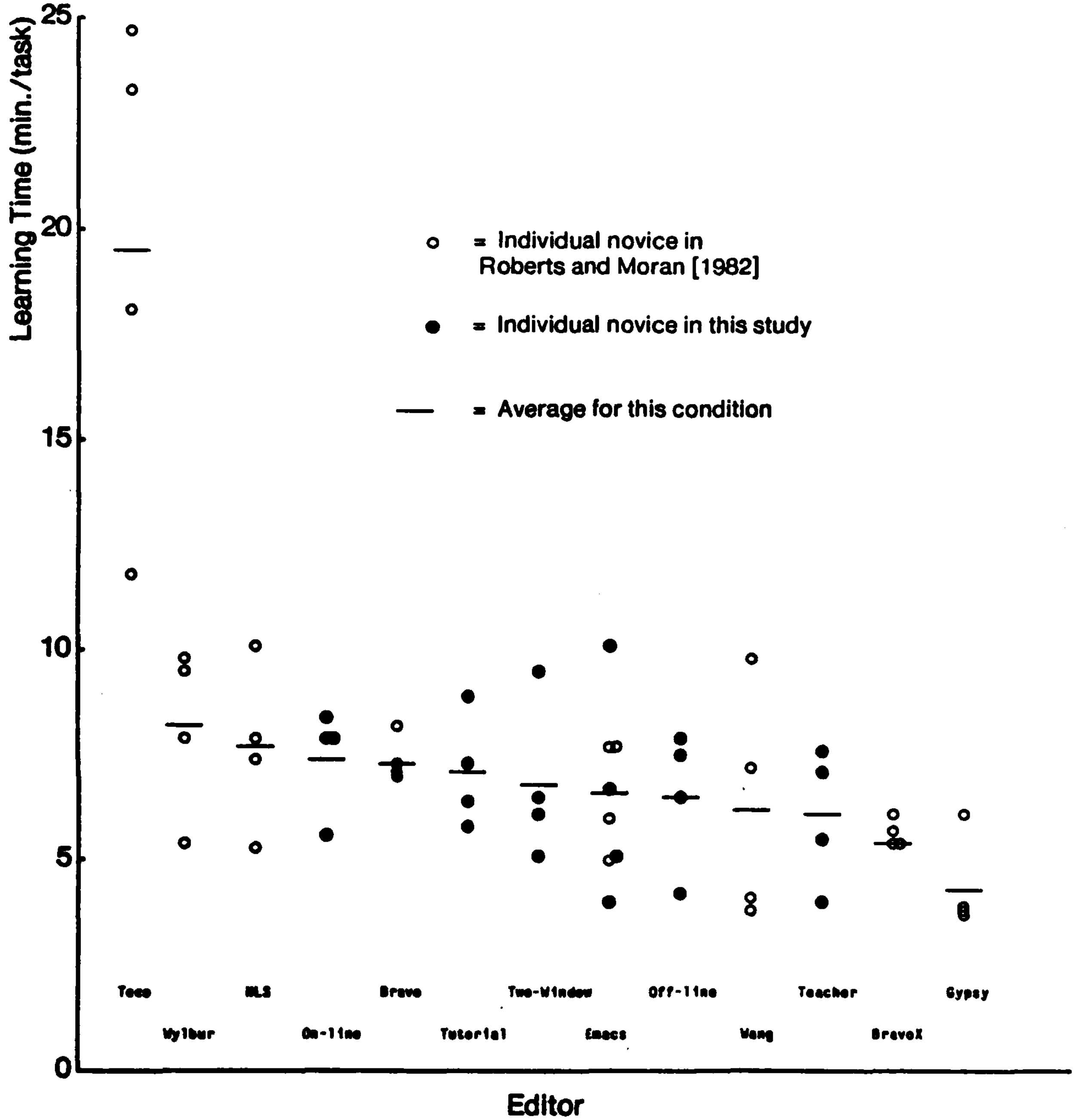


Figure 6-6: Learning scores, ZED compared with other editors.

7. DISCUSSION

7.1. Overall Results

The primary result of the experiment is that all of the ZED teaching techniques produced the same learning in approximately the same amount of time. Further, learning scores in each condition were scattered, with no real outliers. The results of the statistical comparison and the scatter of scores indicates that increasing the number of users is unlikely to change the result; if significant differences are found they will be small. This effectively answers our basic question of which teaching technique to recommend for beginning ZOG users: to our surprise, our teaching techniques are equivalent.

It is important for this result that the users did learn to use ZED effectively. They did not become experts, of course, but did go from zero knowledge of ZED to basic editing at a rate better than one-third that of experts doing similar tasks. [See our study of ZED experts (Robertson, C. Kamila, McCracken & Newell, 1981).] The fact that the retention test scores a week later were essentially equivalent to those of Quiz 5 indicates that their learning was solid. Interviews with the users after the experiment indicated that previous knowledge of the very different editor EMACS neither helped nor hindered their learning in any obvious way.

The overall result is important because each of the techniques provides a viable path for real users to learn ZOG editing (and each technique has been so used outside the experimental situation). ZOG editing, along with basic ZOG searching (learned by the users in Part I), provides an almost complete introduction to ZOG (the minor missing element being how to evoke parameterized procedures). The ZOG system itself provides a complete management information system and interface. Thus, any of these techniques can be used with roughly equivalent results, either as a function of user preference or other considerations.

In particular, this study shows that a self-contained ZOG, with the various forms of on-line documentation, can be just as effective for the novice as ZOG with a teacher or with hard copy documentation. For example, in using ZOG on the USS CARL VINSON, one-to-one instruction is largely unworkable. Classroom instruction is the norm, even though it tends to separate instruction from hands-on experience with the system. It is of value to know that these users can learn on their own, as they use the system, without the immediate attention of a teacher.

Clearly, these results are at odds with our expectations and with the traditional view of such teaching materials. It is highly unlikely that somehow our assignment of users to conditions could have masked a difference. As a check, our users in the different experimental conditions showed no significant difference in learning ZOG (Part I) and there is no correlation between learning ZOG and learning ZED on an individual basis.

7.2. Explaining the Equivalence of the Techniques

How can the virtual equivalence of the teaching techniques be explained? Consider a simple view of learning as the acquisition of an amount of knowledge in some effective organization. This ignores the effects of practice (Newell & Rosenbloom, 1981). However, it will do for this experiment, where the user continually moves on to new material, thus dealing with all aspects to roughly the same, relatively light, extent. Then the time to learn can be factored into two components:

`Learning-time = Time-for-volume + Time-for-complexities.`

This acknowledges two effects. First, the more material there is to be learned, the more time it will take. This should be roughly proportional to the volume of material [as in constant reading rates or the total-time learning law (Cooper & Pantle, 1967)]. Second, difficulties and confusions can arise that add (perhaps substantial) learning time. Such times are independent of the volume of material. In particular, interference effects, which are known to be a strong component of learning from classical learning experiments, are lumped with the complexities, rather than being part of the volume component. In this situation, what is to be learned is highly organized, so in general the material is conceptually separated and interference occurs only in a few places, analogous to other difficulties.

In terms of this simple view, the learning time is constant over the teaching conditions for two reasons. First, learning time is governed primarily by the volume requirement, which is a function of the editor, not the teaching technique. Second, none of the teaching techniques introduced serious unique additional complexities, so that the time due to complexities and errors was uncorrelated with condition. Several lines of evidence can be brought to bear to increase the plausibility of this explanation.

First, Roberts and Moran (1983) showed that a dominant factor in the time to learn the various editors in their experiment (see Figure 6-2) was something they called *procedural complexity*. This was a measure of the amount of knowledge required to encode the methods used in the tasks involved in the quizzes. The exact definition of procedural complexity is not critical here; it is indicative of the number of memory chunks (as that term is commonly used in cognitive psychology) that the user requires to encode the specific methods, each such chunk being the result of learning.[5] Hence, the procedural complexity is a measure of the volume of material to be learned. Roberts and Moran show that some alternative measures of the complexity of the editor (e.g., number of core commands and the number of physical operations) account for substantially less of the learning time. Thus, there is some evidence that the time to learn basic editing is a reflection of the volume component. And of course in our experiment, the editor (ZED) is constant, so that the volume requirement is constant over conditions.

Second, there is some indirect evidence from the way the teaching techniques were produced that all of them were good exemplars of their kind. The text of the *on-line*, *off-line*, and *two-window* manuals, which were the same, had been written by someone with experience in writing. They were extensively edited and rewritten during the preparation of the experiment, with several iterative reviews by another person experienced in technical writing. Care was taken to make it especially clear what the user was to type, and to give examples. This was influenced by previous experiments in which ZED beginners had interpreted (in verbal protocols) the meaning of the instructions in various forms of *on-line* ZED help.

The *tutorial* was the end product of a year-long iterative experiment. Beginners each went through the *tutorial* and evaluated it orally. Overall time to get through the material was observed, plus understandability, and the results of editing tests. We analyzed the results from two users, improved the *tutorial* in response to their problems, ran another two users, and so on, leading to considerable evolution. In particular, the scheme of exercises, done on the instruction frame, evolved. Overall, users went from long, frustrating experiences with inconsistent results, to consistent, reliable coverage of basic material in a period of two to three hours.

The human-teacher technique was taken directly from Robert's original scheme, where it had been used to teach a number of editors by more than one teacher. Thus, the basic approach had evolved to a satisfactory state. Perhaps most important, the basic structure of the Robert's teaching scheme is one of rapidly alternating lessons and exercises in a situation where performance on the editor provides rapid, natural and relatively clear knowledge of results. Certainly these features are known to be ingredients of good teaching techniques. This basic structure was common to all the teaching techniques, as it formed the basic experimental design.

Therefore, the content of the manuals and the *tutorial* was fairly well honed to provide adequate access to knowledge of ZED commands and their proper use. We had built into their structure the responses to many users' questions. They could well be comparable to the *human-teacher* paradigm, which is inherently oriented

---

[5] They counted the number of mental preparation operators (M's) in the encoding of the specific methods according to the Keystroke Model (Card, Moran & Newell, 1983). These are taken to mark the user retrieving or deciding on the next immediate sequence of actions to be performed, each such sequence therefore being an integrated chunk.

to respond to users' questions as they arise. In support of this, though all users could ask questions as needed, the non-teacher users asked an insignificant number of questions. Thus, it is plausible that all the teaching techniques had been brought up to a high enough standard so that they did not introduce idiosyncratic complications (e.g., not being clear about some aspect of the editor) that would have been reflected in learning time due to complexities.

Third, the teaching techniques were each used in different ways — so that the conditions did make a difference in behavior. For instance, the pattern of use shows a trade-off between the number of accesses and the length of time spent in an access (Figure 7-1[6]). The particular variations make reasonable sense. The *tutorial* mandated a large number of short accesses. The *off-line* manual could be left open at a particular page to which the user could refer very briefly. The *on-line* and *two-window* manuals tended to longer trips because the user had to start at the top of the net every time. The *teacher* condition is perhaps less predictable, since the interaction could be pursued to whatever seemed personally appropriate.

Despite this variation, the overall time of access by each technique was basically equal. There are no significant differences in the total time of access. Thus, the appropriate trade-off curve is the number-of-accesses X time-of-access = constant, which would plot as a negative-45^{o} straight line in the logarithms of the measures (as we have done in the figure).[7] This is just what would be expected if what determined behavior was the content of the material to be learned and not the technique that was being used to access it. Thus, this lends additional support to the general explanation being advanced. We have also plotted the point for the *Emacs* condition, which is also a human teacher situation; although it fits the trade-off, it is somewhat separated from the (ZED) *teacher* point.

In sum, if the explanation be accepted that the learning times reflected essentially the volume requirement, we have shown that all of the various techniques for the user to get access to the knowledge about using an editor can be brought to an adequate level so they do not impede the learning. From an experimental point of view this is perhaps a form of ceiling effect, but from an applied point of view it is a highly positive result.

7.3. Field Trial of Two of the Techniques

To establish whether the above conclusion will hold in another environment, two of the teaching techniques (*human teacher* and *two-window*) were taken to the USS CARL VINSON. A procedure was set up to teach the ship's ZOG beginners, using the same stimuli and the same teaching sequence as in our experiment. This was a field situation in which we had no experimental control. However, the goal was to test the teaching techniques in a real-use environment.

For the *human teacher* condition, the experimenter ran Part I and II with six users. The users were officers with varying amounts of familiarity with computing but no or almost no experience with ZOG. Results of the *teacher* condition were as follows. Data was incomplete on three of the users. The other three had an average learning score of 6.40 minutes per task (CV = .31), which does not differ statistically from the scores of users in the five conditions of the main experiment. This shows that the *teacher* condition, at least, does transfer well to that population and that environment. These three users averaged approximately 25 minutes for Part I (ZOG searching instruction).

The ship personnel were to provide teaching with the *two-window* manual. Initial data returned was incomplete, but it does indicate that the users learned to use ZOG/ZED (since several of them did complete

---

[6]The plot is generally significant. The number of distinct accesses of *tutorial* was greater than *on-line*, *two-window*, and *teacher*; and *off-line* was greater than *on-line* and *two-window* (all significant at α = .05 or better). For length of access, *on-line* and *teacher* were each greater than *tutorial* and *off-line* (significant at α = .05).

[7]We have plotted the average points for each condition; the result is the same if the individual points are used.

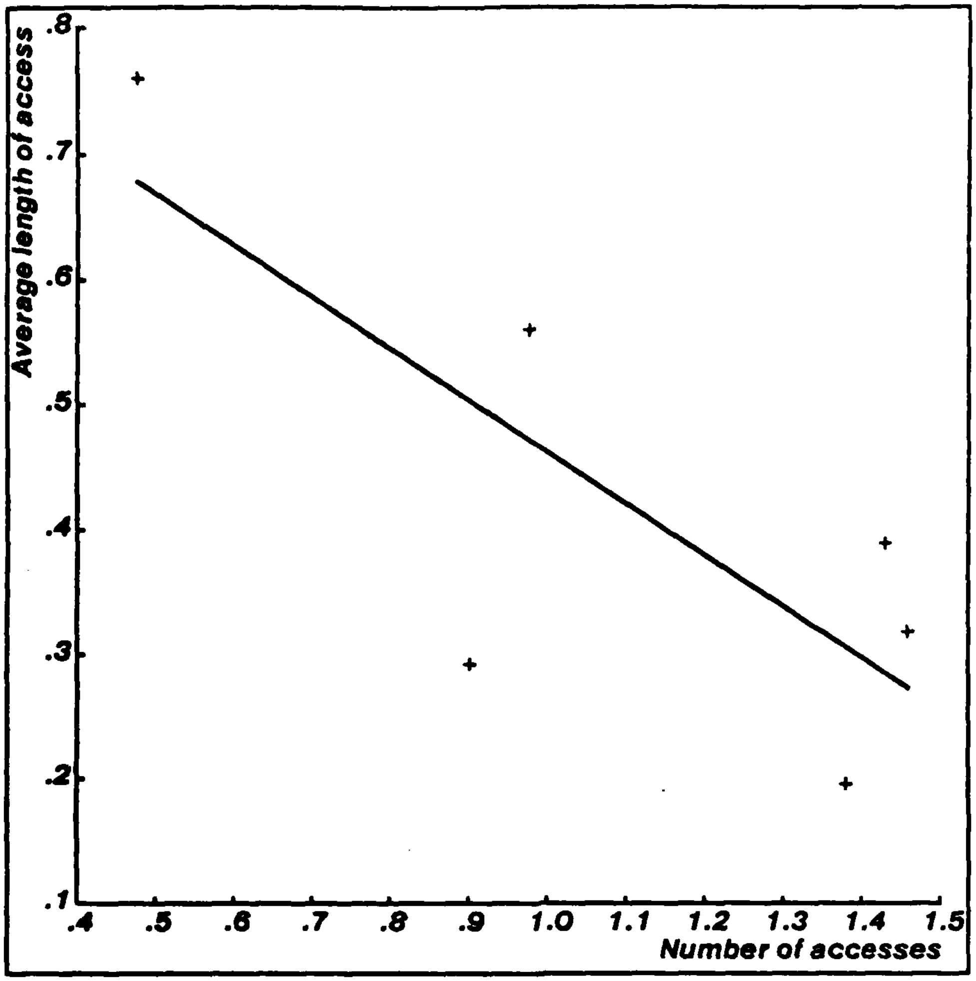


Figure 7-1: Log of number of accesses vs. log of min. per access.

the retention test successfully). Thus the *two-window* teaching technique also shows promise for that environment.

7.4. Basic ZOG Searching

Finally, we have established that basic ZOG searching can be learned in roughly a half hour. Our users enjoyed the ZOG learning (Part I) aspect of the experiment. Several times we had to tell them firmly that their period of ZOG searching practice was over. (This was a game, which was to search a house for clues to a riddle -- one room to a frame.) Some of these users, not finding the answer to the riddle in time, asked to come back later and play the game, or just play with ZOG.

8. CONCLUSIONS

We have found Roberts' method an effective one for measuring ZED learning. We have found that ZED falls in the middle of the range of the editors so far studied. And contrary to the obvious, we have found that all of the teaching techniques we explored are roughly equivalent in the time to learn effectively the basics of the ZED editor. They may all be used with confidence depending on situation and preference.

This experiment was not designed to shed light on these teaching techniques independent of the particular system we used (ZED and ZOG). However, the entire pattern of results, including the embedding of our results within those of Roberts and Moran, and the explanation in terms of the volume component of learning, suggests the conclusions may apply more broadly to learning other editor-like procedural systems.

9. ACKNOWLEDGEMENTS

We would like to acknowledge Rob Akscyn's participation on the initial part of this study (Robertson, C. Kamila & Akscyn, 1982), and the help of Rob Akscyn, Elise Yoder and Don McCracken throughout. We would like especially to thank Sandy Esch for her help with running the experiments and doing the data analysis.

References

Card, S., Moran, T. P., & Newell, A. *The Psychology of Human-Computer Interaction.* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983.

Cooper, E. H., & Pantle, A. J. The total-time hypothesis in verbal learning. *Psychological Bulletin*, 1967, *68*, 221-234.

Newell, A. & Rosenbloom, P. Mechanisms of skill acquisition. In Anderson, J. R. (Ed.), *Cognitive Skills and their Acquisition.* Hillsdale, NJ: Lawrence E. Erlbaum, Associates, 1981.

Newell, A., McCracken, D. L., Robertson, G., & Akscyn, R. ZOG and the USS CARL VINSON. *Computer Science Research Review, 1980-81*, 1982, pp. 95-118. Pittsburgh, PA: Carnegie-Mellon Univ. Department of Computer Science.

Roberts, T. L. *Evaluation of Computer Text Editors* (Tech. Rep. SSL-79-9). Xerox Palo Alto Research Center, November 1979.

Roberts, T L., & Moran, T. P. Evaluation of Text Editors. In *Human Factors in Computer Systems.* Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, 1982. Gaithersburg, MD, March 15-17, 1982.

Roberts, T. L., & Moran, T. P. The Evaluation of Text Editors: Methodology and Empirical Results. *Communications of the ACM*, 1983, *26, No. 4*, 265-283.

Robertson, C. Kamila, & Akscyn, R. Experimental Evaluation of Tools for Teaching the ZOG Frame Editor. In *Man-Machine Systems.* Institute of Electrical Engineers, 1982. University of Manchester Institute of Science and Technology, Manchester, England, July 6-9, 1982.

Robertson, C. Kamila, McCracken, D. L., & Newell, A. Experimental Evaluation of the ZOG Frame Editor. In *Proceedings of the Annual Conference of the Canadian Man-Computer Communications Society.* Canadian Man-Computer Communications Society, 1981. Waterloo, Ontario, Canada, June 10-12, 1981.

Robertson, G., McCracken, D. L., & Newell, A. The ZOG approach to man-machine communication. *International Journal of Man-Machine Studies*, 1981, *14*, 461-488.

Yoder, E., & Akscyn, R. *ZOG Users Guide* (Tech. Rep.). Computer Science Dept., Carnegie-Mellon University, 1982.